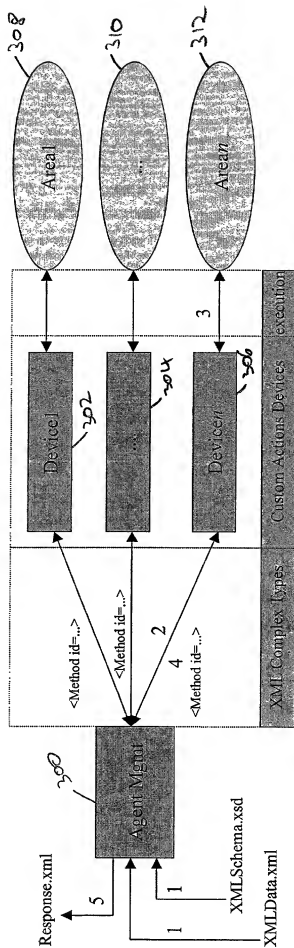


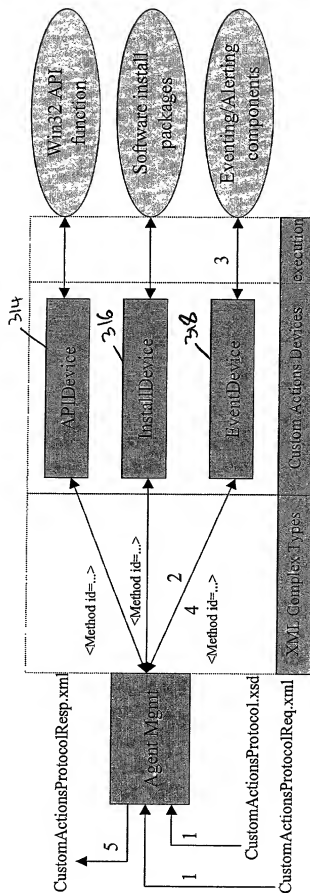
Architecture overview



- 1) Request XML file is passed to Agent Mgmt containing complex types
Agent Mgmt splits the data into parts and hands the `<Method>` over to the corresponding Custom Actions Devices
- 2) The complex type `<Method>` describes the method to execute and its parameters
- 3) Custom Actions Devices execute the functions in a certain area they are responsible for and pack the result into XML complex types
- 4) Complex types containing result data are returned to Agent Mgmt
- 5) XML stream has been packed and is returned

Fig. 1

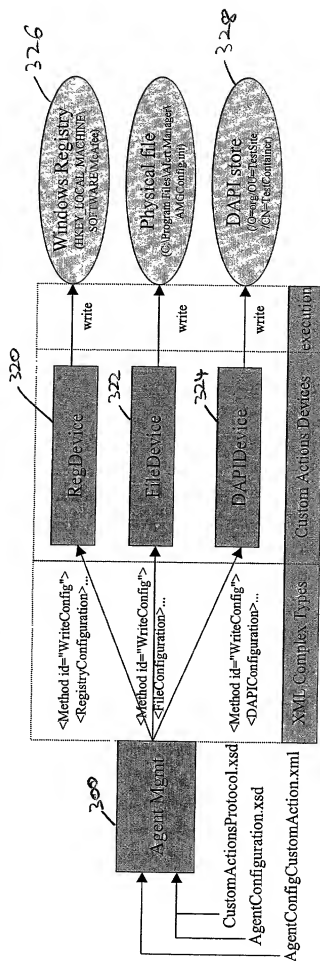
Executing Custom Actions



- 1) Request XML file is passed to Agent Mgmt containing complex types
- 2) The complex type <Method> describes the method to execute and its parameters
Each <Method> is passed to the corresponding Custom Actions Device
- 3) Custom Actions Devices execute the functions and pack the result into XML complex types
- 4) Complex types containing result data are returned to Agent Mgmt
- 5) XML stream has been packed and is returned

Fig. 2

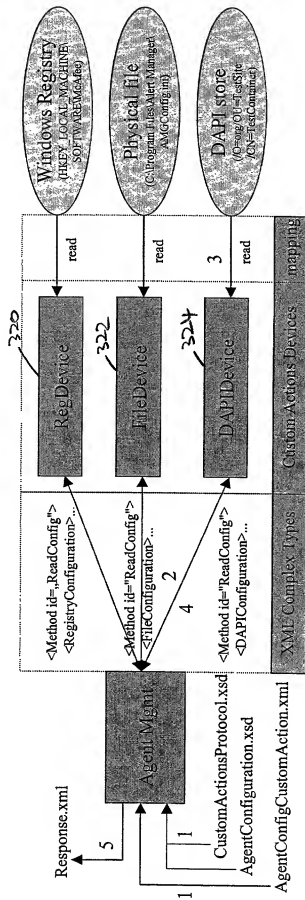
Deploying Configuration Data



- 1) Agent Mgmt receives AgentConfigCustomAction.xml
- 2) .xml file is validated against .xsd file(s) to make sure valid data is written
- 3) XML Complex types are sent to Custom Actions Devices, the parameters containing the configuration data
- 4) Custom Actions Devices update the config store they are responsible for
- 5) Optionally a return value can be returned as a Response.xml file

Fig. 3

Retrieving Configuration Data



- 1) Request XML file is passed to Agent Mgmt containing empty requested complex types
- 2) The empty complex types are passed to the Custom Actions Device
- 3) Complex types are filled by mapping required data from store
- 4) Complex types containing data are returned to Agent Mgmt
- 5) XML stream has been packed and is returned

Fig. 4

Initiator

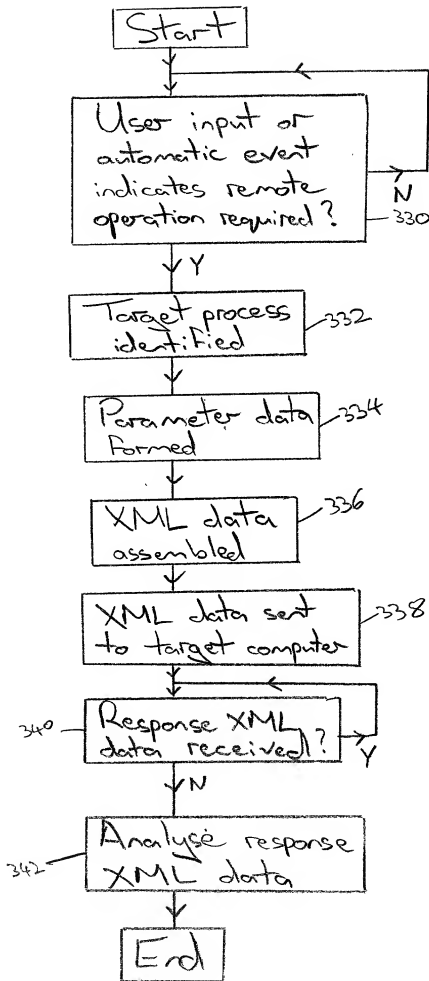


Fig. 5

10091435.030702
20200514T0001

Agent

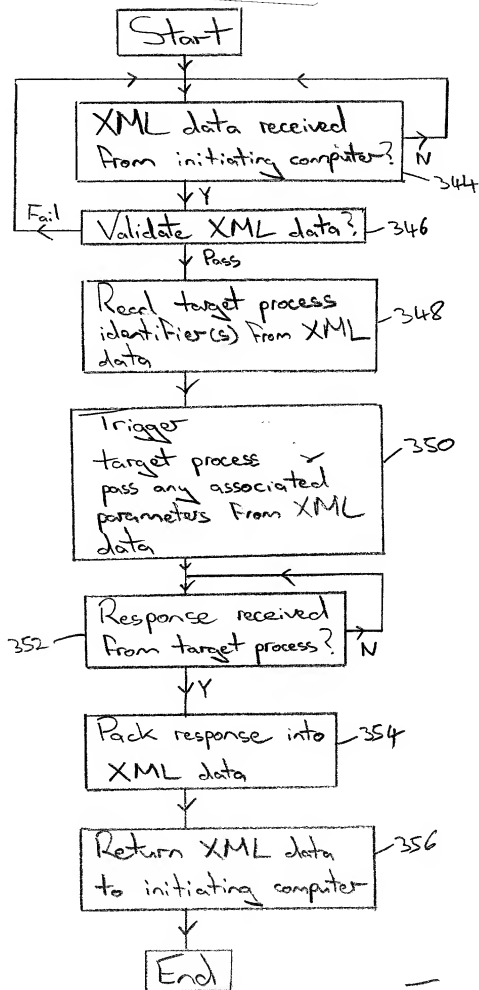
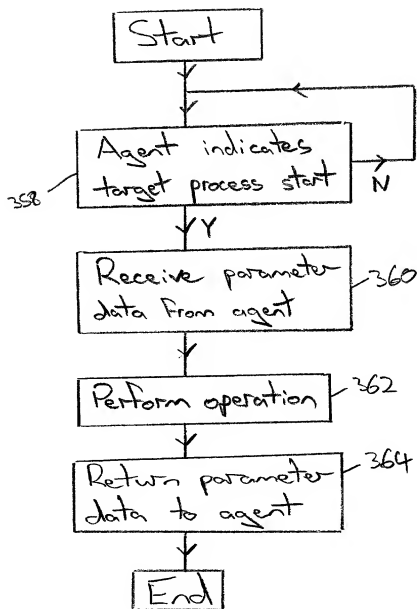


Fig. 6

10091415.030702

Target



10091415.030702

Fig. 7

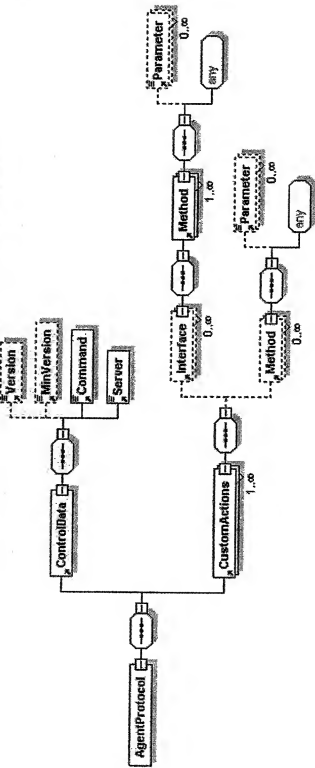
8
 1.01

2020010300702

Protocol Specification	
<ControlData>	contains Agent specific control data like version information, the command to execute and computer information sending the custom actions information.
<Command>	the command field could be used (or easily determine if data is going to be retrieved (RequestCustomAction) or returned after the action has been executed (RespondToCustomAction) optional
<Server>	computer information of the sender. If the Agent is able to process requests in parallel and asynchronously this information is useful. The simplest form of this field contains the computer name
<CustomActions>	any number of custom action COM servers or DLLs or executables required for the tasks are listed within the CustomActions
id	the id specifies a CLSID. If the custom action method is contained in a COM server of the path to a DLL or in a library or an executable implements the method to execute. If a CLSID is specified, the following <Interface> complex type is required to specify the interface of the COM server containing the method to execute. Otherwise <Interface> is not required and any number of <Method> types follow immediately.
<Interface>	only required if the <Method> is implemented in a COM server
id	the interface identifier of the COM server (IID)
<Method>	any number of Methods implemented in the custom action device
id	the Method name. In case of a COM server, this is the method name of the COM interface, else this denotes the name of an exported function or a command line parameter of an executable
<Parameter>	any number of parameters required for the method. This includes requested output parameters, which are listed, but don't contain data and input parameters which have a different value on response than on request
id	the name of the parameter
type	can be any standard XML datatype
input	possible values are 'in', 'out' and 'input'. Specifies if the parameter is requested, passed to the function or passed to the function for modification.
<any>	any other non standard XML datatype can follow

Fig. 9

2020ED-57416001



CustomActionsProtocol.xsd

CustomActionsProtocolReg.xml

Inventory: NEDBAL, M, et al
SN unknown/Sheet 10 of 27
Atty. Dkt.: 550-321

```
<?xml version="1.0" ?>
- <AgentProtocol xmlns="http://www.nai.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.nai.com CustomActionsProtocol.xsd">
- <ControlData>
  <Version>0x01000001</Version>
  <MinVersion>0x01000001</MinVersion>
  <Command>RequestCustomAction</Command>
  <Server>ned1wnts2ke</Server>
</ControlData>
- <CustomActions
  id="<AGENT_INSTALLED_DIR>\\CustomActionsLibrary\\CustAct1.dll">
- <Method id="GetRegStringValue">
  <Parameter id="Key" type="xs:string"
    inout="in"><AGENT_INSTALLED_REGKEY></Parameter>
  <Parameter id="Valuename" type="xs:string"
    inout="in">AgentVersion</Parameter>
  <Parameter id="Result" type="xs:string" inout="out" />
</Method>
</CustomActions>
- <CustomActions id="{06E0062A-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="ExecuteSilentInstallation">
    <Parameter id="ProductName" type="xs:string"
      inout="in">TestInstallProduct</Parameter>
    <Parameter id="ProductVersion" type="xs:decimal"
      inout="in">0x01000001</Parameter>
    <Parameter id="Location" type="xs:string"
      inout="in">c:\InstallImages</Parameter>
    <Parameter id="Result" type="xs:string" inout="out" />
  </Method>
</Interface>
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="GetSystemDirectory">
    <Parameter id="Directory" type="xs:string" inout="out" />
    <Parameter id="Result" type="xs:decimal" inout="out" />
  </Method>
</Interface>
</CustomActions>
- <CustomActions id="{06E0062B-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{A000CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="TriggerEvent">
    <Parameter id="EventID" type="xs:decimal"
      inout="in">1000</Parameter>
    <Parameter id="EventDescription" type="xs:decimal"
      inout="in">The event %EventID% has been triggered by %
      USERNAME% on computer %COMPUTERNAME%. The %
      FILENAME% file is infected with %VIRUSNAME%. This has
      been detected by engineversion %ENGINEVERSION%
      datversion %DATVERSION%.</Parameter>
    <Parameter id="COMPUTERNAME" type="xs:string"
      inout="in">sourcecomputer</Parameter>
    <Parameter id="USERNAME" type="xs:string"
      inout="in">sourceuser</Parameter>
    <Parameter id="FILENAME" type="xs:string"
      inout="in">kernel32.dll</Parameter>
    <Parameter id="VIRUSNAME" type="xs:string"
```

Fig. 10A

10091415.030702

Custom Actions Protocol Req. xml

Inventor: NEDBAL, M. et al
SN unknown/Sheet 11 of 27
Atty. Dkt.: 550-321

```
inout="in">Nimdda</Parameter>
  <Parameter id="ENGINEVERSION" type="xs:decimal"
    inout="in">0x04005001</Parameter>
  <Parameter id="DATVERSION" type="xs:decimal"
    inout="in">0x07003009</Parameter>
  <Parameter id="Result" type="xs:string" inout="out" />
</Method>
</Interface>
</CustomActions>
</AgentProtocol>
```

10091415.030702

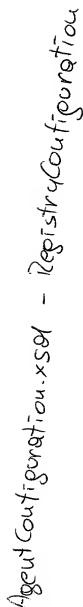
Fig. 10B

```

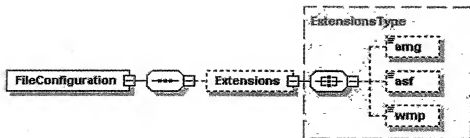
<?xml version="1.0" ?>
- <AgentProtocol xmlns="http://www.nai.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.nai.com CustomActionsProtocol.xsd">
- <ControlData>
  <Version>0x01000001</Version>
  <MinVersion>0x01000001</MinVersion>
  <Command>RespondToCustomAction</Command>
  <Server>ned1wnts2ke</Server>
</ControlData>
- <CustomActions
  id="<AGENT_INSTALLED_DIR>\\CustomActionsLibrary\\CustAct1.dll">
- <Method id="GetRegStringValue">
  <Parameter id="Result" type="xs:string"
    inout="out">5.0.1.10</Parameter>
</Method>
</CustomActions>
- <CustomActions id="{06E0062A-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="ExecuteSilentInstallation">
    <Parameter id="Result" type="xs:string" inout="out">Error: Invalid
      Image path specified.</Parameter>
  </Method>
  </Interface>
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="GetSystemDirectory">
    <Parameter id="Directory" type="xs:string"
      inout="out">C:\Winnt\System32</Parameter>
    <Parameter id="Result" type="xs:decimal"
      inout="out">0</Parameter>
  </Method>
  </Interface>
</CustomActions>
- <CustomActions id="{06E0062B-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{A000CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="TriggerEvent">
    <Parameter id="Result" type="xs:string" inout="out">Event sent to
      testcomputer2</Parameter>
  </Method>
  </Interface>
</CustomActions>
</AgentProtocol>

```

Fig. 11



12
LiOH



AgentConfiguration.xsd - FileConfiguration

10091445.030702

Fig. 13



AgentConfiguration.xsd - DAPIConfiguration

10091415.030702

Fig. 14

AgentConfigCustomAction.xml

inventor: NEDBAL, M. et al
SN unknown/Sheet 16 of 27
Atty. Dkt.: 550-321

```
<?xml version="1.0" ?>
- <AgentProtocol xmlns="http://www.nai.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.nai.com CustomActionsProtocol.xsd
  http://www.nai.com AgentConfiguration.xsd">
- <ControlData>
  <Version>0x01000001</Version>
  <MinVersion>0x01000001</MinVersion>
  <Command>RequestCustomAction</Command>
  <Server>ned1wnts2ke</Server>
</ControlData>
- <CustomActions id="RegistryMapping.dll">
- <Method id="WriteConfig">
- <RegistryConfiguration
  id="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee">
- <Product id="Alert Manager">
  <Version>0x04070000</Version>
  <DisplayName>Alert Manager 4.7</DisplayName>
- <Language id="0407">
  <Version>0x01000002</Version>
- <Event id="1">
  <LONGDESCRIPT>Das ist eine Test-Nachricht von Alert
  Manager.</LONGDESCRIPT>
  <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
  <Severity>5</Severity>
  <Enabled>1</Enabled>
</Event>
</Language>
- <Language id="0409">
  <Version>0x01000002</Version>
- <Event id="1">
  <LONGDESCRIPT>This is an alert manager test
  message.</LONGDESCRIPT>
  <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
  <Severity>0</Severity>
  <Enabled>1</Enabled>
</Event>
- <Event id="2">
  <LONGDESCRIPT>Text of event 2.</LONGDESCRIPT>
  <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
  <Severity>1</Severity>
</Event>
</Language>
</Product>
</RegistryConfiguration>
</Method>
- <Method id="ReadConfig">
  <RegistryConfiguration
    id="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\*" />
</Method>
</CustomActions>
- <CustomActions id="INIFileMapping.dll">
- <Method id="WriteConfig">
- <FileConfiguration id="C:\Program Files\Alert
  Manager\AMGConfig.ini">
- <Extensions>
```

Fig. 15A

10091415.030702

AgentConfigCustomAction.xml

Inventor: NEDBAL, M. et al
SN unknown/Sheet 17 of 27
Atty. Dkt.: 550-321

```
<amg>AMGConfig</amg>
<asf>MPEGVideo</asf>
<wmp>MPEGVideo2</wmp>
</Extensions>
</FileConfiguration>
</Method>
- <Method id="ReadConfig">
  <FileConfiguration id="C:\Program Files\Alert
    Manager\AMGConfig.ini" />
  </Method>
</CustomActions>
- <CustomActions id="MAPIMapping.dll">
- <Method id="WriteConfig">
  - <DAPIConfiguration id="/O=org/OU=TestSite/CN=TestContainer">
    <BinaryProperty>0123456789ABCDEF00000</BinaryProperty>
  </DAPIConfiguration>
  </Method>
- <Method id="ReadConfig">
  <DAPIConfiguration id="/O=org/OU=TestSite/CN=TestContainer" />
  </Method>
</CustomActions>
</AgentProtocol>
```

10091415.030702

Fig. 15B

Source

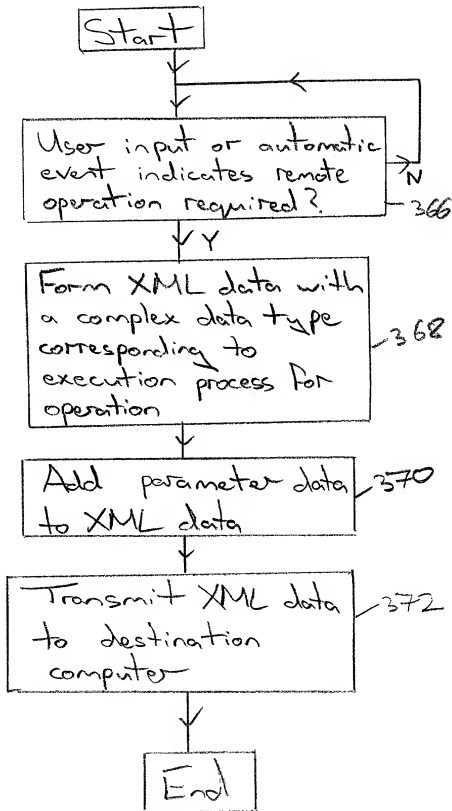


Fig. 16

10091415.030702

Destination

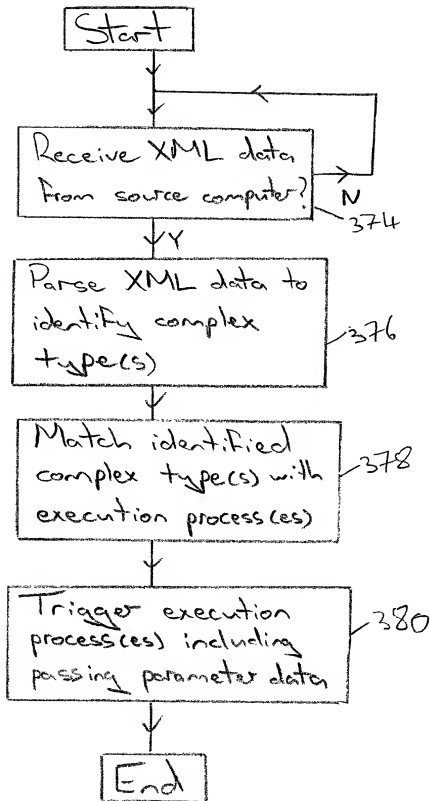
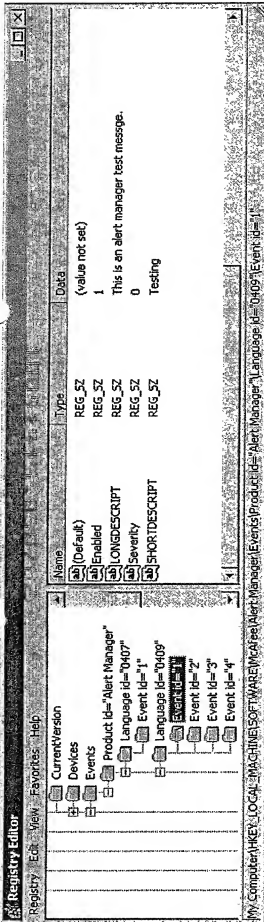
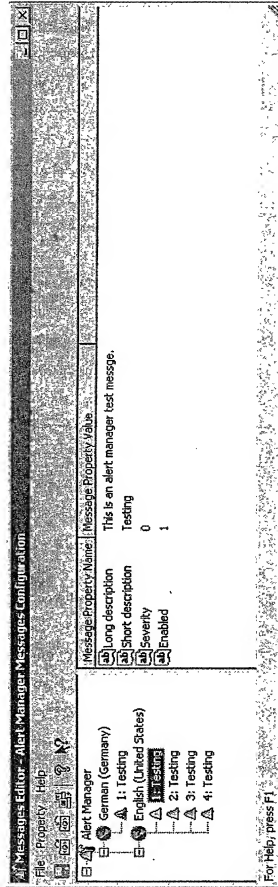


Fig. 17

10091445.030702



Registry Data Fig. 18



DOM Data View Fig. 19

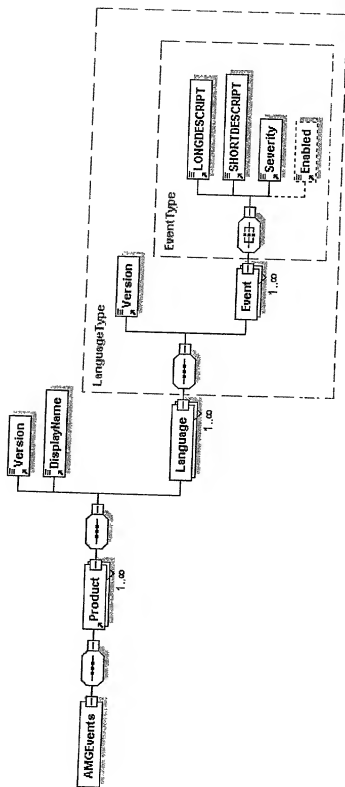
<?xml version="1.0" ?>
- <AMGEvents xmlns="http://www.nai.com"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.nai.com AMGEvents.xsd">
 - <Product id="Alert Manager">
 <Version>0x04070000</Version>
 <DisplayName>Alert Manager 4.7</DisplayName>
 - <Language id="0407">
 <Version>0x01000002</Version>
 - <Event id="1">
 <LONGDESCRIPTION>Das ist eine Test-Nachricht von Alert
 Manager.</LONGDESCRIPTION>
 <SHORTDESCRIPTION>Testing</SHORTDESCRIPTION>
 <Severity>5</Severity>
 <Enabled>1</Enabled>
 </Event>
 </Language>
 - <Language id="0409">
 <Version>0x01000002</Version>
 - <Event id="1">
 <LONGDESCRIPTION>This is an alert manager test
 message.</LONGDESCRIPTION>
 <SHORTDESCRIPTION>Testing</SHORTDESCRIPTION>
 <Severity>0</Severity>
 <Enabled>1</Enabled>
 </Event>
 - <Event id="2">
 <LONGDESCRIPTION>Text of event 2.</LONGDESCRIPTION>
 <SHORTDESCRIPTION>Testing</SHORTDESCRIPTION>
 <Severity>1</Severity>
 </Event>
 - <Event id="3">
 <LONGDESCRIPTION>Text of event 3.</LONGDESCRIPTION>
 <SHORTDESCRIPTION>Testing</SHORTDESCRIPTION>
 <Severity>1</Severity>
 </Event>
 - <Event id="4">
 <LONGDESCRIPTION>Text of event 4.</LONGDESCRIPTION>
 <SHORTDESCRIPTION>Testing</SHORTDESCRIPTION>
 <Severity>1</Severity>
 </Event>
 </Language>
 </Product>
</AMGEvents>

XML Data

Fig. 20

10091415.030702

207020-574T602T 10091415-030702



Generated with XMLSpy Schema Editor www.xmlspy.com

XSD Data

Fig. 21

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XML Spy v4.0.2 U (http://www.xmlspy.com) by Napalm
{Napalm} -->
- <xs:schema targetNamespace="http://www.nai.com"
  xmlns="http://www.nai.com"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="DisplayName" type="xs:string" />
  <xs:element name="Enabled" type="xs:boolean" />
  - <xs:complexType name="EventType">
    - <xs:all>
      <xs:element ref="LONGDESCRIPT" />
      <xs:element ref="SHORTDESCRIPT" />
      <xs:element ref="Severity" />
      <xs:element ref="Enabled" minOccurs="0" />
    </xs:all>
    <xs:attribute name="id" type="xs:string" use="required" />
  </xs:complexType>
- <xs:complexType name="LanguageType">
  - <xs:sequence>
    <xs:element ref="Version" />
    <xs:element name="Event" type="EventType"
      maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required" />
</xs:complexType>
- <xs:element name="Product">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element ref="Version" />
      <xs:element ref="DisplayName" />
      <xs:element name="Language" type="LanguageType"
        maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required" />
  </xs:complexType>
</xs:element>
- <xs:element name="AMGEvents">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element ref="Product" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="LONGDESCRIPT" type="xs:string" />
<xs:element name="SHORTDESCRIPT" type="xs:string" />
<xs:element name="Severity" type="xs:string" />
<xs:element name="Version" type="xs:string" />
</xs:schema>
```

XSD Data

Fig. 22

10091415.030702

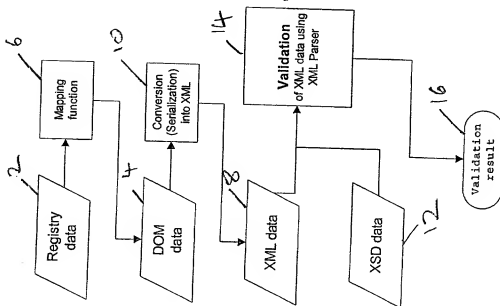


Fig. 23

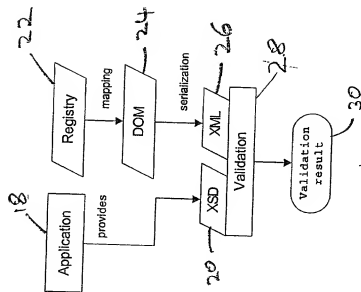


Fig. 24

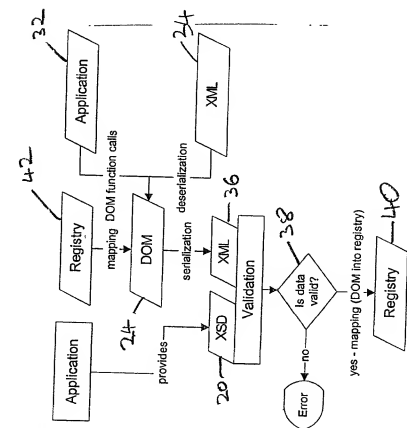


Fig. 25

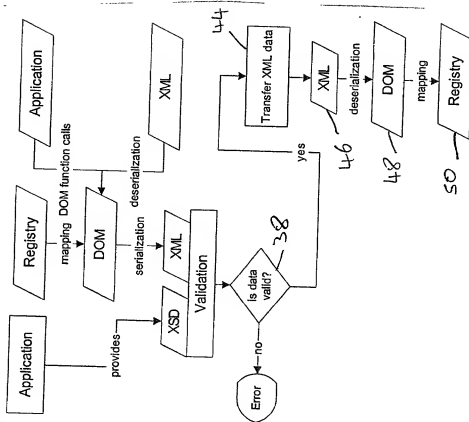


Fig. 26

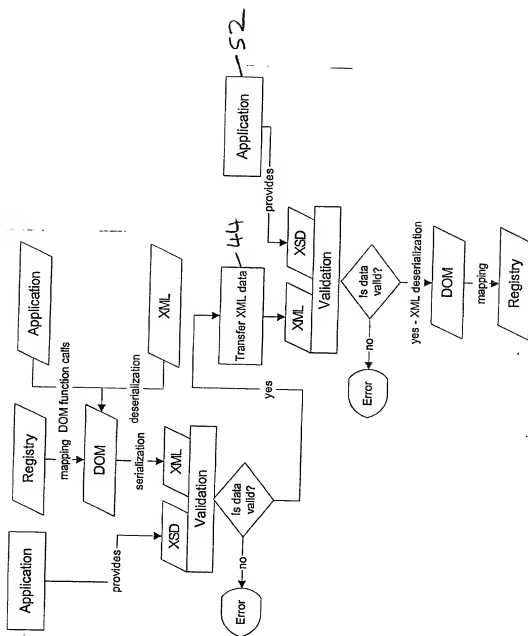


Fig. 27

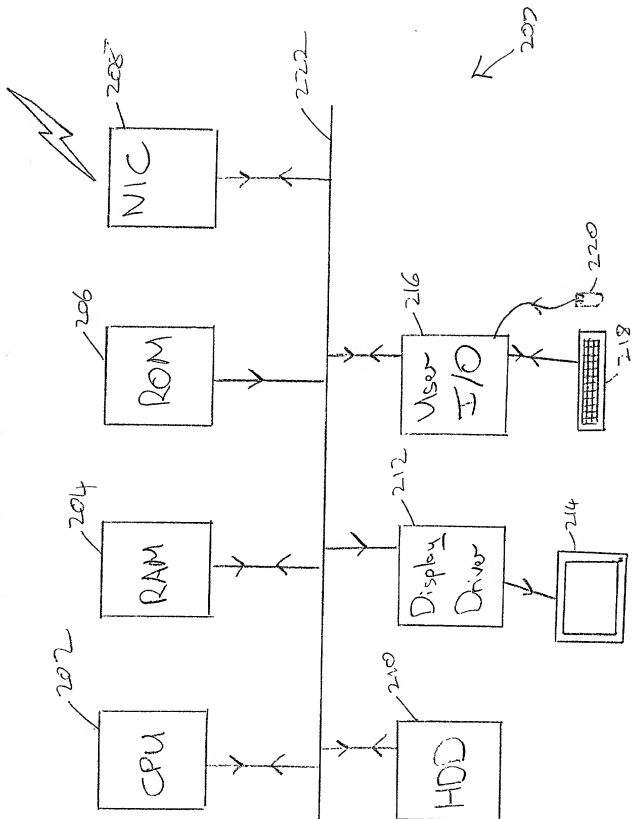


Fig. 28